Analysis Algorithm Kohonen and Momentum on the Back propagation Neural Network

Purwa Hasan Putra Muhammad Zarlis, H Mawengkang

***Abstract**: In this paper, it is faster to achieve the target error with the weight addition of Kohonen and Momentum combination in Back propagation when compared with the addition of random weight. The error quadratic decrease in Back propagation method training with the target error of 0.007 with random weight reaches the target error at 55 iterations, whereas the weight addition of Kohonen and Momentum in Back propagation reaches the target error at 36 iterations. The test results with the weight addition of Kohonen and Momentum combination in Back propagation is better when compared with the addition of random weight, where it can recognize the test data reaching the accuracy of 96.53%.*

***Index terms**: Neural Network, Kohonen, Momentum, Back propagation*

I.INTRODUCION

Air temperature is the average kinetic energy size of molecular movement. Temperature is a non linier condition in 24 hours can change. Artificial Neural Networks have become very important over the past few years. The neural network methodology technique for solving non-linear problems is very high [1].

Kohonen can be trained in a short period of time with a few optimization techniques such as "winning" neurons search scope limit [2, 3]. While momentum method is used to accelerate the training process in recognizing data patterns. In this research, the author is going to optimize Back propagation method by adding Kohonen and Momentum algorithm in recognizing patterns temperature in Medan. The author is investigate Kohonen and Momentum algorithm can reduce training time in recognizing the pattern of temperature data provided.

II. RELATED RESEARCH

As input patterns are presented to ANN networks, output patterns are produced. ANN's neurons constitute layers: the output layer, one or more hidden layers, and the input layer. Hence, information flows to the output layer from the input layer via the hidden layer, with the latter forming a platform for input-output layer association (Meier & Rix, 1994).One of the findings in the study by Nazzal and Tatari (2013), who used ANN for back propagation of flexible system moduli, it was observed that ANN exhibits the capability of predicting layer moduli values of systems with success; with Kohonen-enabled field deflection measurements on focus. Similarly, it was noted that ANN adoption yields a significant reduction in computation time while simplifying the back propagation process. These findings concurred with those established by Meier and Rix (1995), who affirmed that from the perspective of speed, ANN models' rapid prediction ability imply that they are able to analyze

100,000 Kohonen deflection profiles in mini and micro seconds. The eventuality is the provision of a significant advantage to system engineers who end up assessing transportation infrastructure conditions non-destructively and in real time (Saltan & Terzi, 2008). Despite this promising nature of ANN to foster the back propagation of flexible system moduli in real time, Goktepe, Agar and Lav (2005) cautioned that a number of considerations need to be made. For instance, it was suggested that the accuracy of ANN-led outcomes is dependent on the integrity and quality of Kohonen deflection data that the operators collect from the field. Hence, the need to define Kohonen reporting requirements, data analysis approach, and testing requirements while seeking to assure ANN outcome accuracy cannot be overemphasized. Accurate information regarding layer thickness (at the testing points of Kohonen) has also been documented to play a crucial role in steering successful back propagation of the stiffness of layers or predict maximum deflections, strains, and stresses, which constitute critical system responses (Saltan, Uz & Aktas, 2013). The implication is that the degree of accuracy of the Kohonen data obtained at the testing points determines the level of success achieved in ANN-led back propagation of flexible system moduli.

To provide system sustainability, accurate maintenance strategies are deemed important to transportation agencies. With imposed loadings aiding in measuring surface deflections via Kohonen, the resultant

Revised Manuscript Received on December 22, 2018.

Purwa Hasan Putra, Graduate school of Computer Science, Universitas Sumatera Utara,Malaysia
Muhammad Zarlis, Graduate school of Computer Science, Universitas Sumatera Utara, Malaysia
H Mawengkang,Graduate School of Mathematics, Universitas Sumatera Utara, Malaysia Email: kabuhocan@gmail.com

non-destructive system evaluation has proved critical in informing these agencies based on the resultant structural assessments. Hence, the criticality of system layer back propagation to determine properties using programs such as ANN cannot be overemphasized. Figure 2 represents a back-propagation neural network, the principles on which back propagation via ANN works. From the figure, BP and AO indicate error back-propagation and directions of activation while *o1* to *o2* and $i_1$to $i_4$depict the problem's output and input variables. It is also worth noting that the hidden layers' neurons are depicted by $h_{11}$to h_{23}. In figure 3, ANN's typical processing unit is represented. In the figure, X_i refers to the input signal while W_{ij}constitutes the connection weights. Indeed, the input signal is meant for prior neurons of *N* number. Similarly, $y_i$entails the output signal, $net_i$the net input signal, and θ_i the activation threshold (Seo, Kim, Cho & Jeong, 2013).

In such a case, each interval of about 500 iterations is expected to be accompanied by an examination of the decrease in the error trend, with the error criterion MAE aiding in the evaluation of approximations between the computed EANN and the target moduli Etarget [28]. this step in the ANN design and training procedure yields a great enhancement of chances of achieving the adopted convergence criteria. Given that a variety of layer moduli combinations could still yield similar errors, the process of deflection basin verification is deemed ideal in fostering the selection of a set of moduli perceived to be more adequate; also achieved via ANN. Whereas target outputs are represented by the measured deflection basins, inputs are defined by the computed moduli EANN. Hence, the optimal solution becomes the option with better regression analysis indicators, as well as the least error between field-measured and predicted deflection basins. Therefore, it can be inferred that the forecasting reliability of ANN options is determined by the desirable extent to which their resultant computations tend to match field-measured deflection basins. To achieve the matching accuracy, it is expected that the measured and the computed deflections exhibit a linear regression[11].

Upon establishing the optimum ANN, it becomes important to assess the generalization capability[12]. this assessment is achieved via data sets different from those utilized in designing and training the network. From the perspective of verification, the predicted layer moduli values are compared with the actual condition of the system; considering further the structural defects and material variability of the selected stretch. In situations where spatial variations of the respective moduli suggest that estimations of material such as asphalt concrete are the least, a notable attribute that could be linked to such a trend constitutes asphalt base's lower stiffness. In a quest to establish potential reasons behind the atypical or low values of the resultant layer moduli, inclusions of the actual condition of the system surface are imperative. As documented by [13]. the surface condition affects the estimated moduli values. In stretches closer to areas with severe distresses or higher rutting and deflections, lower values are obtained [14]. On the other hand, deteriorated regions prompting special initiatives via maintenance procedures are associated with lower moduli (Goktepe, Agar & Lav, 2006).

Given that ANN fosters real-time back propagation in relation to simultaneous data acquisition and analysis, the emerging benefit is that its application aids in alleviating both the indirect costs of commuting delays and the direct costs of traffic control. Hence, work crews who use ANN as a layer moduli back propagation tool are likely to be at less risk as they do not necessarily need to work in the middle of traffic lanes, outside their vehicles. However, the need for crew workers to consider sub-grade thickness during the back propagation process cannot be overemphasized. Overall, the paper has established that the evolution of ANN as a tool for backcalculating flexible system moduli from Kohonen and Momentum data promises outcomes such as speed and efficiency. In future, the use of ANN as a tool for the back propagation of Kohonen and Momentum-generated data is poised to stretch beyond the merits of speed and efficiency to foster cost-effectiveness from the commuting and traffic control-related expenses; yet quality is also predicted to remain uncompromised.

III.PROPOSED METHOD

A.Algorithm Kohonen

Kohonen network, as shown in is a two layer feed-forward network. Learning of KNN is hybrid, since from input to hidden layer unsupervised learning and from hidden to output layer supervised learning is used. Theorem 1 and theorem 2, shows that the basis function of Kohonen network is universal approximator. Training algorithm of KNN is given below [7]:

1. Select initial weights Wij random values from input vector range and learning rate λ € =[0,1]
2. Apply step 3-7, upto stopping criteria is false, stopping criteria may be number of iteration or learning parameter is sufficently small (say €).
3. Apply euclidean measure from input vector and weight vector, for j=1...,m
   $D\ (j) = \sum_{i=1}^{m} \sum_{j=1}^{m} (xi - wij)2$
4. **Calculate winning node say index J, so that D(j) is minimum**
5. **J within a specified neighbourhood of j and for all i, calculate new weights as in equation**

Update new weigt using

$$\textbf{Wij (new)} = \left(1 - \lambda\right) \boldsymbol{Wij(old)} + \lambda \boldsymbol{Xi}$$

Update new weight λ using

$$\lambda\ (t + 1) = 0.5 * \lambda\ (t)$$

Calculate error and test stopping criteria of network

B .Neural Network

From the direction of activation propagation, the new neuron is reached by input signals emerging from prior processing units. The unit evaluates these signals in relation to their respective connection weights before multiplying thee individual input signals by their corresponding connection weights. In so doing, internal activity of neurons is calculated in relation to the input signals' weighted summation (Sharma & Das, 2008).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication
1350

To calculate the net input signal, the equation used holds:

$$net_i = \sum_{j=1}^{N}(W_{ij}X_j) - \theta_i$$

In situations where the resultant input signal is more than the threshold limit value, the response of the neuron is expected to indicate $y_i$based on selected transfer neurons

(over linear and threshold transfer). With changes in y_i (the output signal value) for the sigmoidal function perceived to lie between 0 and 1, the resultant equation holds:

$$f(x) = \frac{1}{(1+e^{-x})}$$

Indeed the back-propagation learning rule while employing ANN for back propagation lies in the quest to reduce the error or difference between the calculated and the desired output values; translating into supervised learning. As the learning or training process begins, randomly initialized connection weights are provided before the updating stage based on the level of the error. As each of the steps employed in forward propagation ends, an objective function is applied towards the calculation of the error E_k. In this case,

$$E^k = \frac{1}{2}\sum_i [t_i^k - y_i^k]^2$$

Notably, neuron *k* and *i* data's actual output is represented by t_i^k. As observed above, the calculated error determines the adjustment done to the connection weights (ARA Inc. & ERES Consultants Division, 2004). Similarly, expressing ΔW_{ij} (the amount of change between *j* and *i* neurons) requires calculations of derivatives of error terms in relation to the resultant connection weights. Calculating this change gives:

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = -\eta \sum_k \left(\frac{\partial E^k}{\partial W_{ij}}\right)$$

The learning coefficient, which is expected to exceed zero, is represented by η. An application of the chain rule gives

$$\frac{\partial E^k}{\partial W_{ij}}$$

Rewriting in the delta term (δ_i^k) gives:

$$-\frac{\partial E^k}{\partial W_{ij}} = -\frac{\partial E^k}{\partial y_i}\frac{\partial y_i}{\partial net_i}\frac{\partial net_i}{\partial W_{ij}} = -\delta_i^k \frac{\partial net_i}{\partial W_{ij}} = -\delta_i^k X_j$$

Given the availability of the estimated and actual output signals, the computation of the delta term in the output layer can be done. Regarding the hidden layer, the output signals expected to be sent are unknown. As such, the role of the delta term δ_m^k in this case lies in the calculation of the current delta value. Imperative to highlight is that the latter value employs neurons *m*; with the previous *i*-th layer forming its location (Ceylan, Guclu, functions *f(x)*. In response to the net internal activity, the output response gives:

$y_i = f(net_i)$

Regarding the resultant functions, they fall into three categories namely: sigmoidal, threshold, and linear. Indeed, the sigmoidal transfer function exhibits more similarity when compared to real

Tutumluer & Thompson, 2005). Expressing the resultant general delta rule gives:

$$\delta_i^k = \begin{cases} (t_i^k - y_i^k) f'(net_i^k) & \text{for output layers} \\ \sum_m \delta_m^k W_{im} f'(net_i^k) & \text{for hidden layers} \end{cases}$$

The sigmoidal function's derivative becomes:

$f'(x) = f(x)\{1 - f(x)\}$

Upon completing these activation propagation steps, the process of back-propagation commences from output layers to the input layers. This process, as observed by Chatti, Ji and Harichandran (2004), is marked by adjustments of link weights in the respective iterations, which are successive. The eventuality is that the activation direction's outputs become the back-propagation direction's inputs (Goel & Das, 2008). For iterations that follow, updating new connection weights of *j* and *i* neurons gives:

$$W_{ij}(it+1) = W_{ij}(it) + \eta \sum_k \delta_i^k X_j^k + \alpha[W_{ij}(it) - W_{ij}(it-1)]$$

Indeed, the momentum term is represented by α and considers changes in weight in the preceding iterations meant to prevent the trapping in of local minimums by the algorithm and, also, to yield oscillations. The modification of bias values is also done in the same manner as the link weights to give:

$$\theta_i(it+1) = \theta_i(it) + \eta \sum_k \delta_i^k + \alpha[\theta_i(it) - \theta_i(it-1)]$$

Indeed, the above steps are expected to be repeated for the individual data in training sets in an iterative manner while seeking to achieve the minimum error between the calculated and the desired output. Given that ANN exhibits the capacity to solve complex problems that are resource-intensive in an accurate and fast manner, the trend accounts for its extensive application in system problems [15]. Indeed, the adaptive back propagation approach had its initial applications spearheaded by Meier and Rix (1994), targeting attributes such as dynamic deflection basins, layer properties, and surface wave inversions. Notably, an achievement of correct backcalculated layer moduli demands accurate deflection measurements. However[16] . observed that it is unlikely to be realistic to expect an exact match between the systematic deflections responsible for neural network training and the experimental deflections. Hence, a number of measurement errors, which are equipment-related, arise. On the one hand, random errors constitute the noise or random measurement errors that pose the difficulty of reproduction [18][19] . On the other hand, systematic errors are repeatable and, through proper measurement apparatus calibration, they can be minimized. Universally, the Kohonen test specifications hold that that the repeatability error does not exceed 0.08 mils while the respective geophones' systematic errors do not exceed 2%[17].

The development of ANN as an intelligent software is attributed to the convergence of learning and adaptation. Through the latter processes, variable and complex systems are modeled. According to Gopalakrishnan and Thompson (2004), the solving of problems via ANN is dependent on prior knowledge in such a way that new information is incorporation in the entire evolving learning procedure, ensuring that the forecasting capabilities of ANNs are increased. It has also been established that ANN yields algorithmic developments responsible for mimicking the manner in which humans think. In turn, the software treats nonlinear and more complex problems in a manner deemed to be rational, following the development of regression computations' multivariate models. Therefore, it is evident that the human brain's efficiency and complex pattern forms an inspiration of the ANN technique. Specifically, a high degree of connectivity characterizing a number of neurons translates into intelligence. As the neurons of ANN transmit, process, and receive information or signals to related neurons to which it is connected, the resultant and individual links exhibit unique and associated values referred to as weights [20] . As these weights are fitted, they end up simulating certain behaviors or features.

C.Back Propagation

It is also worth noting that the outcomes obtained after applying ANN to establish the layer moduli of flexible systems is dependent on the nature of interconnection or architecture between neurons. In a related observation [21] documented that the ANN modeling outcomes are also determined by the strength values characterizing weight values (or connections). The implication is that multi-dimensional and non-linear problems are more likely to arise in situations where complex structures are presence. In the study [22]. IT was asserted that ANN system's typical elements include transfer functions, input functions, error functions, and the learning rules. Hence, the need to establish definitions of these elements via trial-and-error procedures cannot be overemphasized. Imperative to note further is that the ANN development procedure constitutes two leading steps. During the training stage, input-output sets of data enhance the learning process. In situations where the desired outputs are known [23].documented that there is a need to reinforce the learning process. However, situations where, for the specified outputs, the desired outputs are given, it was observed that supervision needs to be embraced in relation to the learning process [24]. Apart from the training stage, the model development of ANN exhibits the testing procedure. Indeed, the importance of this stage lies in the decision to predict ANN's capacity to produce reasonable outputs in relation to any new data. The implication is that the testing data is responsible for reporting the model's performance statistics in its entirety [25]. Upon establishing desirable outcomes in the two stages (ANN training and testing), the model produced is poised to exhibit the capability of steering reliable predictions when presented with data sets that are unknown.

Back propagation is a supervised learning algorithm and is commonly used by perceptrons with multiple layers to change the weights associated with neurons in the hidden layer. The back propagation training algorithm basically consists of 3 stages, namely [9] :

a. Input the value of training data to obtain the output value.
b. Back propagation of the error value obtained.
c. Adjust the connection weight to minimize the error value.

All three stages are repeated continuously to get the desired error value. After the training is done.

D.Momentum Back propagation

Some of the types of systems that could characterize such a system include a four-layer system and a three-layer system. Whereas the former could feature a sub-base whose stiffness exceeds that of the granular base, the latter could be characterized by layers whose stiffness decreases with depth. To apply the ANN back propagation procedure, the initial process is expected to entail the design and training practice using the available database of the system. The next step is expected to constitute an assessment of forecasting capabilities. Lastly, the proposed model is verified via the establishment of comparisons between the system's actual condition and the layer moduli values predicted.

a. Initialize weights (take the initial weight with fairly small random value), Epoh = 1 and MSE = 1.
b. Determine theEpoh Maximum, Learning Rate (α), and Target Error.
c. Perform the following 4 to 12 steps during (Epoh <Epoh maximum) and (MSE > Target Error).
d. Epoh = Epoh + 1.
e. Feedforward
   1) Each of input unit(X_1, $i=1,2,3,\ldots,n$) receives the signal xi and forwads the signal to all unit on the layer above it (hidden layer).
   2) Each of hidden unit(Z_1, $j=1,2,3,\ldots,p$)sums the weight of the input signal, shown by the equation (1).

      $$z\_in_j = b_{1j} + \sum_{i=1}^{n} x_i\, v_{ij}$$

      Then use the activation function to calculate the output signal, shown by the equation (2).

      $$z_j = f(z_{in_j})$$

      And send those signals to all unit in the top layer (output layer)
   3) Each of output unit (YK, K=1,2,3,..,m) sums weighted input signals, shown by the equation (3).

$$y_{in_k} = w_{0k} + \sum_{i=1}^{p} z_i\, w_{jk}$$

Then use the activation function to calculate the output signal, shown by the equation (4).

$$y_k = f(y\_in_k)$$

f. Feedback (Momentum Back propagation)

1) Each of output unit*(Y_k, k=1,2,3,…,m)*receives the target pattern asthe training input pattern, then calculate the error, shown by the equation (5).

$$\delta_k = (t_k - y_k) f'(y_{in_k})$$

$\varphi 2_{jk} = \delta_k - z_j$

$\beta 2_k = 8_k$

Then calculate the weight correction (which will later be used to correct Wjk value), shown by the equation (6).

$$\Delta w_{jk} = \alpha \varphi 2_{jk} + \mu \Delta w_{jk}$$

Performed as many as the number of hidden layers to the previous hidden layer.

2) Each of hidden unit (Zj, j=1,2,3,..,p) sums the input delta (from units located on the top layer), shown by the equation (7).

$$\delta\_in_j = \sum_{k=1}^{m} \delta 2_k\, w_{jk}$$

Multiply this by a derivative of the activation error function, shown by the equation (8).

$$\delta_j = \delta_{in_j} f'\left(z_{in_j}\right)$$

$\varphi 1_{ij} = \delta 1_j - x_j$

$\beta 1_j - \delta 1_j$

Then calculate the weight correlation (which will later be used to fix Vij value), shownby the equation (9).

$$\Delta v_{ij} = \alpha \varphi_{ij} + \mu \Delta v_{ij}$$

Calculate the bias correction also (which will later be used to improve the value of bij), shown by the equation (10).

$$\Delta b1_j = \alpha \varphi 1_j + \mu \Delta b1_j$$

μ is a constant of momentum with a range [0.1].

g. Improvement weight

1) Each of output unit (Yk, k=1,2,3,..,m) fixes its bias and weights (j=0,1,2,..p), shown by the equation (11).

$$w_{jk}(new) = w_{jk}(long) + \Delta w_{jk}$$

$$b2_k(new) = b2_k(long) + \Delta b2_k$$

2) Each of hidden unit (Zj, j=1,2,3,..p) fixes its bias and weights (i=0,1,2,..n), shown by the equation (12).

$$v_{ij}(new) = v_{ij}(long) + \Delta v_{ij}$$

$$b1_j(new) = b1_j(long) + \Delta b1_j$$

3) Calculate thevalue of MSE, shown by the equation (13).

$$\text{MSE} = \frac{\sum \text{tk} - \text{Yk}}{\text{n}}$$

IV.RESULTS AND DISCUSSION

In this research, the pattern recognition of Medan city temperature data would be done using artificial neural network Back propagation method that is done by dividing into 2 parts, namely: data for training, data for testing. Data used in Medan city temperature is the data from 1998-2017.

To know whether the addition of Kohonen and Momentum algorithm with Back propagation method, then some tests were done. The first test the researcher did the training with the number of hidden 6, alpha 1, target error differed between 0.01, 0.0095, 0.009, 0.008, 0.007, max epoh 50, and momentum 0.5.

Before it is processed the data was normalized first. The normalization of the data was done in order that the network output was in accordance with the activation function used. The data was normalized in the interval [0, 1] because the data used is positive or 0. It is also related to the activation function given, that is the binary sigmoid.

Sigmoid function is asymptotic function (never reach 0 or 1), then data transformation was done at smaller interval [0,1; 0.8], shown by the equation (16).

$$x' = \frac{0.8(x-a)}{b-a} + 0.1$$

*a*is the minimum data, *b*is the maximum, *x*is the data to be normalized, and x' is the data that has been transformed.

To achieve this process, it is important to establish the leading variables shaping the nature of input-output data sets [26].documented that system behavior is affected by numerous variables. As such, an identification of variables deemed to be the most influential is appropriate and this procedure can be achieved via sensitivity analyses. In the selected example, some of the most influential variables could include surface system deflections in the respective tests, layer thickness, and the load level. It is also worth noting that for the respective layer materials, typical values can be assumed in relation to Poisson's ratio. These layer materials include the lower layer, stabilized sub-base, granular base, and the asphalt layer [27]. Figure 5 illustrations this sample procedure depicting the back propagation model. The ANN design and training process culminates into deflection basin verification. The latter, which constitutes sensitivity studies, holds that the transfer function represents sigmoid, the input function represents the dot product, and the pre-processing defined by the mean standard deviation. The results of the trainingcan be seen at table 1.

Table 1. Training Result

#	Error (epoch)				
	0.01	0.0095	0.009	0.008	0.007
B	30	33	37	44	55
B + K	29	32	36	39	55
B + M (0.5)	24	(1 5)26	27	30	37
B + K + M	22	23	24	28	36

B = Back propagation
K = Kohonen
M= Momentum

In Table 1 we can see the results of research with the addition of Kohonen and Momentum combination weight in Back propagation achieved the target error faaster when compared with the addition of random weight. Quadratic decrease error in Back propagation method training with target error 0.007can be seen at figure 1.

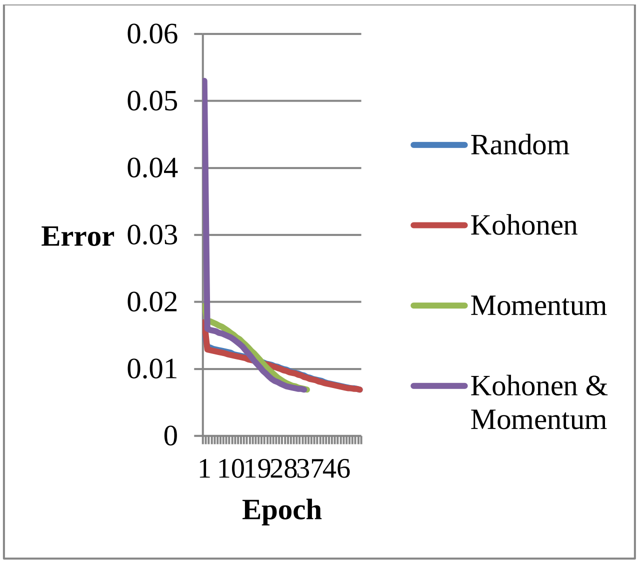

Figure 1. QuadrateDecrease Error

In Figure 1 we can see a decrease in the quadrate of error by the addition of Kohonen and Momentum weight in Back propagation can reach the target error 0.007 at 36 iterations, whereas Back propagation with random weight reaches target error of 0.007 at 55 iterations. Furthermore, the data were tested using Back propagation of random weight and Back propagation with the addition of Kohonen and Momentum weight. The result of the testcan be seen at table 2.

Table 2. Testing Result

#	Error (%)				
	0.01	0.0095	0.009	0.008	0.007
B	95.32	95.42	95.63	95.88	96.38
B + K + M	95.85	95.87	96.02	96.32	96.53

B = Back propagation
K = Kohonen
M = Momentum

In Table 2 we can see the results of the test with the addition of Kohonen and Momentum combination weight in Back propagation is better when compared with the addition of random weight. Back propagation with the addition of Kohonen and Momentum weights can recognize the test data reaching the accuracy of 96.53%.

REFERENCES

1 Ihwan, Andi, Artificial Neural Network Method Back propagation For Monthly Rainfall Estimation In Ketapang West, *Proceed Semirata FMIPA University Of Lampun*, 2013.

2 Ahmad A and Ismail M N, *Clustering the Imbalanced Datasets using Modified KohonenSelf-Organizing Map (KSOM).* Computing Conference 2017, pp 751-755.

3 Khvorostukhina E et al. 2017 *Performance Improvements of a Kohonen Self-OrganizingTraining Algorithm.* pp 456-458.

4 Akashdeep, et al., Time Series Analysis Of Forecasting Indian Rainfall, *International Journal Of Inventive Engineering And Sciences (IJIES): ISSN 2319-9598, Volume-1, Issue-6, May 2013.*

5 Priya, et al., Time Series Analysis Of Forecasting Indian Rainfall, *International Journal Of Innovatios & Advancement In Computer Science (IJIACS), Volume 3, Issue 1, April*, 2014

6 Singh U P, Tiwari A, et al. 1989,*Kohonen Neural Network Model References For Nonliniear Discrete Time Systems, 3rd IEEE International Conference on "Computational Intelligence and Communication Technology.*

7 Jaiswal J K, Das R, Application Of Artificial Neural Networks With Back propagation Technique In The Financial Data,*IOP Conf. Series: Materials Science and Engineering*, 2017.

8 Sutojo, T., et al, 2010, *Artificial Intelligence*, Yogyakarta: Andi Offset.

9 Izhari F, Dhany H W, Zarlis M, et al, Analysis Back propagation Methods With Neural Network For Prediction Of Children's Ability In Psychomotoric, *2nd International Conference on Computing and Applied Informatics*, 2017.

10 Nurcahaya, Pradana, T.P., et al,Utilization Of Speed Region Growing Segmentation And Back propagation Neural Network Momentum For Classification Of White Blooed Cell Types, 2013.

11.ARA Inc. & ERES Consultants Division (2004). *Guide for mechanistic-empirical design of new and rehabilitated pavement structures*, Final Report, Part 2. National Cooperative Highway Research Program-NCHRP, TRB, NRC

12. Ceylan, H., Guclu, A., Tutumluer, E., & Thompson, M. (2005). Back propagation of full-depth asphalt pavement layer moduli considering nonlinear stress-dependent subgrade behavior. *International Journal of Pavement Engineering*, *6*(3), 171-82

13. Chatti, K., Ji, Y., & Harichandran, R. (2004). *Dynamic time domain back propagation of layer moduli, damping, and thicknesses in flexible pavements* (pp. 106-116). Michigan: Transportation Research Record

14. Goel, A., & Das, A. (2008). Non-destructive testing of asphalt pavements for structural condition evaluation: a state of the art. *Journal of Non Destructive Testing and Evaluation*, *23*(2), 121-40

15. Goktepe, A., Agar, E., & Lav, A. (2006). Role of learning algorithm in neural network-based back propagation of flexible pavements. *Journal of Computing in Civil Engineering*, *20*(5), 370-373

16. Goktepe, A., Agar, E., & Lav, H. (2005). Advances in back-calculating the mechanical properties of flexible pavements. *Advances in engineering software*, *37*, 421-431

17. Gopalakrishnan, K. (2010). Neural network-swarm intelligence hybrid nonlinear optimization algorithm for pavement moduli back-calculation. *Journal of Transportation Engineering - ASCE*, *136*(6), 528-536

18. Gopalakrishnan, K., & Khaitan, K. (2010). Finite element based adaptative neuro-fuzzy inference technique for parameter identification of multi-layered transportation structures. *Transport*, *25*(1), 58-65

19. Gopalakrishnan, K., & Thompson, M. (2004). Back propagation of airport flexible pavement non-linear moduli using artificial neural networks. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference* 2, 652-57

20. Kim, D., Kim, J., & Mun, S. (2010). Normalisation methods on neural networks for predicting pavement layer moduli. *Road & Transport Research*, *19*(3), 38-46

21. Mehta, Y., & Roque, R. (2003). Evaluation of FWD data for determination of layer moduli of pavements. *Journal of Materials in Civil Engineering (ASCE)*, 25-31

22. Meier, R., & Rix, G. (1994). Back propagation of flexible pavement moduli using artificial neural networks. *Transportation Research Record*, (1448), 72-81

23. Meier, R., & Rix, G. (1995). Back propagation of flexible pavement moduli from dynamic deflection basins using artificial neural networks. *Transportation Research Record*, (1473), 72-81

24. Nazzal, M., & Tatari, O. (2013). Evaluating the use of neural networks and genetic algorithms for prediction of subgrade resilient modulus. *International Journal of Pavement Engineering*, *14*(4), 364-373

25. Saltan, M. & Terzi, S. (2008). Modeling deflection basin using neural networks with cross-validation technique in backcalculating flexible pavement layer moduli. *Advances in Engineering Software*, *39*(7), 588-592
26. Saltan, M., Uz, V., & Aktas, B. (2013). Artificial neural networks-based back propagation of the structural properties of a typical flexible pavement. *Neural Computing & Applications*, *23*(6), 1703-1710
27. Seo, J., Kim, Y., Cho, J., & Jeong, S. (2013). Estimation of in situ dynamic modulus by using MEPDG dynamic modulus and FWD data at different temperatures. *International Journal of Pavement Engineering*, *14*(4), 343-353
28. Sharma, S., & Das, A. (2008). Back-calculation of pavement layer moduli from falling weight deflectometer data, using an artificial neural network. *Canadian Journal of Civil Engineering*, *35*, 57-66